Arnaud DABOUIS
Vincent HURBOURQUE
IMAC 1

PROJET C : Traitement d'images

SOMMAIRE

1) Schéma et structure
- **a) Schéma de la structure**
- **b) Les structures utilisées**
  - *La structure image*
  - *La structure calque*
  - *La structure LUT*
  - *La structure Historique et la structure info_historique*

2) Gestion des fichiers
- **a)Les fonctions images**
  - *Allouer une image*
  - *Désallouer une image*
  - *Créer une nouvelle image*
  - *Lire une image*
  - *Créer une image*
  - Utilisation dans le projet
- **b) Construction de l'histogramme**
  - *Initialiser l'histogramme*
  - *Créer l'histogramme*
  - Utilisation dans le projet
- **c) Les fonctions du calque**
  - *Allouer un calque*
  - *Créer le premier calque*
  - *Créer d'autres calques*
  - *Modifier le paramètre d'opacité*
  - *Modifier la fonction de mélange*
  - *Supprimer un calque*
  - *Naviguer dans les calques*
  - *Charger une image sur un calque*
  - *Appliquer un LUT à un calque*
  - *Afficher un calque*
  - Utilisation dans le projet
  - *Sortie de l'image finale*
- **d) Les luts**
  - *Allouer un lut*
  - *Désallouer un lut*
  - *Créer un lut*
  - *Insérer un lut*
  - *Supprimer un lut*
  - *Les effets possibles*
  - Utilisation dans le projet
- **e) L'historique**
  - *Initialiser une pile*
  - *Tester si elle est vide*
  - *Empiler*
  - *Dépile*
  - *Affichage de l'historique*
  - *Annuler une opération*
  - Utilisation dans le projet
- **f) L'IHM**
  - *L'application des luts directs*
  - *Détection du clavier*
  - *Les modes*
  - *Affichage inverse pour avoir l'image à l'endroit*
- **g) Le main**

Nous avons réalisé une application de traitement d'images permettant le chargement d'images, l'ajout de calques sur ces images et l'application de LUTs sur chacun de ces calques. Elle comprend la possibilité d'afficher un histogramme de l'image de départ et de l'image finale, d'enregistrer l'image finale sous forme d'un fichier .ppm, et d'annuler les dernières opérations effectuées par le biais d'un historique.
Durant tout le déroulement du traitement de l'image, une interface graphique permet d'obtenir un rendu en temps réel des modifications effectuées. En outre, le contrôle de l'application se fait directement au clavier.

Organisation du travail :
En ce qui concerne l'organisation du projet nous nous sommes réparties les diverses tâches :
-Arnaud s'est occupé de l'IHM (lut en ligne de commande, gestion du clavier et des appels aux fonctions,mode) et pour les calques de l'ajout, suppression, navigation et des changements possibles (opacité, opération) sur celui-ci.
-Vincent s'est occupé de créer les structures (fichier all_fonctions.h),des fonctions de gestion d'image (lecture, écriture, copie), de l'historique, des luts, de la sortie de l'image finale avec l'histogramme finale, de l'affichage de l'image courante (plus de la fonction qui inverse l'image) avec les luts.
Nous avons fait le main et l'histogramme ensemble, tout comme ce rapport.
Le mode d'emploi a été écrit par Arnaud, et Vincent a organisé les fichiers et les a commentés.

1) Schéma et structure

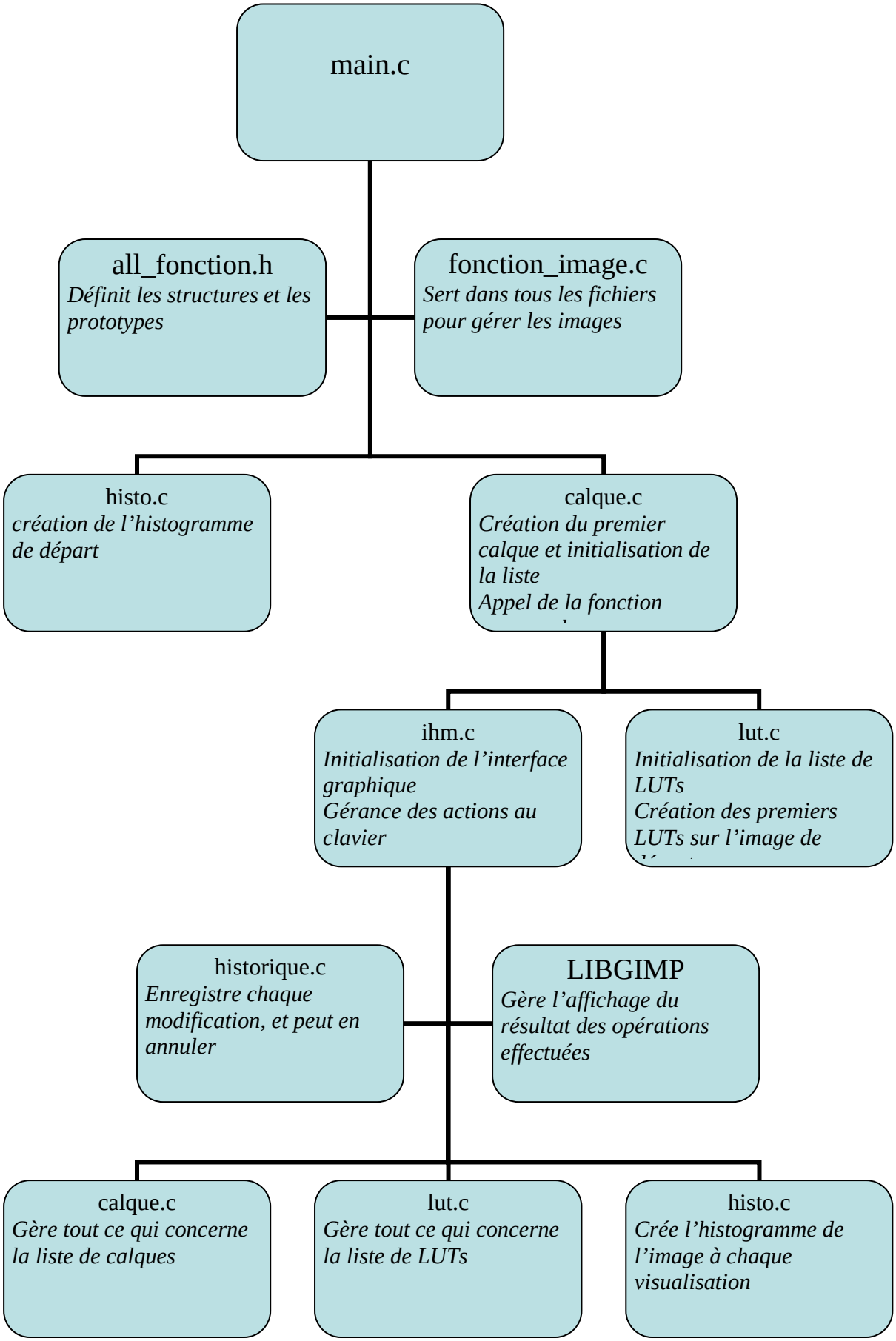
a) Schéma de la structure

Un dessin arborescent montrant les fichiers et leurs liaisons majeures est disponible en page suivante.

b) Les structures utilisées

Nous avons défini les structures nécessaires dans le fichier all_fonctions.h, inclus au début de chaque fichier de type .c. Voyons quelles sont ces structures :

La structure image
Pour cette structure, il a fallu analyser un fichier ppm afin de récolter toutes les informations importantes et nécessaires à sa création. Ainsi la structure comprendra : Un type (P6 ici), la hauteur de l'image, sa largeur, le codage, un tableau de unsigned char contenant l'ensemble des pixels de l'image et enfin 3 tableaux d'entiers contenant chacun la quantité de pixel rouge, vert et bleu (pour l'histogramme).


main.c
all_fonction.h
Définit les structures et les prototypes
fonction_image.c
Sert dans tous les fichiers pour gérer les images
histo.c
création de l'histogramme de départ
calque.c
Création du premier calque et initialisation de la liste
Appel de la fonction
ihm.c
Initialisation de l'interface graphique
Gérance des actions au clavier
lut.c
Initialisation de la liste de LUTs
Création des premiers LUTs sur l'image de
historique.c
Enregistre chaque modification, et peut en annuler
LIBGIMP
Gère l'affichage du résultat des opérations effectuées
calque.c
Gère tout ce qui concerne la liste de calques
lut.c
Gère tout ce qui concerne la liste de LUTs
histo.c
Crée l'histogramme de l'image à chaque visualisation

La structure calque
La structure calque comporte une structure image, car un calque est une image.
De plus, un calque a une opacité (définie par un float) et une opération (0 pour l'addition et 1 pour la multiplication). Il existe un champ appelé id qui permet d'attribuer une identité à chaque calque; ce sera utile dans l'historique pour connaître quelles opérations ont été affectées à quel calque.

Le projet permet de manipuler plusieurs calques, qui seront placés en tant que maillons d'une liste doublement chaînée. Ainsi, chaque structure calque comporte deux pointeurs, un vers le calque suivant et l'autre vers le calque précédent.

La structure lut
Un LUT est un tableau de 255 cases qui permet de savoir comment réagit un pixel face à un effet.
La structure comprend un type qui permet d'identifier le type d'effet (1 pour l'effet inverse par exemple) et elle contient 3 tableaux de unsigned char qui permettent de gérer les effets sur les pixels rouge, vert ou bleu.
Chaque calque pouvant avoir plusieurs LUTs, il a fallu implémenter une liste de LUTs. La structure contient donc aussi 2 pointeurs, un vers le LUT suivant et l'autre vers le LUT précédent.

La structure Historique et la structure info_historique
La structure info_historique est une structure de donnée pure c'est-à-dire qu'elle permet de se souvenir des informations essentielles à l'historique.
On a besoin de connaître l'id du calque, son opacité, le type de lut qui a été supprimé ou ajouté, et le type d'opération (addition ou multiplication) si celui-ci a été changé.
La structure de donnée étant faite, il faut construire notre structure Pile. La pile qui gérera l'historique est un tableau d'info_historique, sa taille max est fixé à 100.La pile contient évidemment un sommet.

Maintenant nous allons analyser comment est construit chaque fichier et quel est son rôle dans le projet.

2) Gestion des fichiers

a)Les fonctions images

Allouer une image
Cette fonction permet d'allouer les tableaux de l'image qui contiendront les pixels de l'image. La taille du tableau contenant tous les pixels doit être largeur*longueur*3.
Les autres tableaux qui permettront de connaître la quantité des rouges, verts et bleus sont des tableau de 255.

Désallouer une image
Il s'agit de simple free sur les 4 tableaux crées.

Créer une nouvelle image
La fonction consiste à copier la structure d'une image et à la mettre dans une autre.
On a ainsi une image de départ, à laquelle l'on prend toutes ses caractéristiques (type, codage, tableau de pixels...) pour les affecter à une image d'arrivée.

Lire une image
C'est l'une des fonctions essentielles du projet. Elle ouvre le fichier ppm qui sera passé en argument; si l'ouverture s'est bien passée, alors on peut procéder à l'extraction des informations importantes du fichier image, c'est-à-dire son type, son codage, les pixels du tableau, etc…. grâce a des scanf et des fgets, et surtout le fread pour le tableau de pixels.
Nous avons supprimé la ligne de commentaire en testant si le premier caractère est un #, dans ce cas nous continuons la lecture.
Le fichier est fermé à la fin de la fonction.

Créer une image
Cette fonction permet de créer un nouveau fichier et d'écrire grâce aux fonctions fprintf et fwrite les informations essentielles, c'est-à-dire : le type, la largeur et la hauteur, le codage et tout le tableau de unsigned char.
Le fichier est fermé à la fin de la fonction.

Utilisation dans le projet
On lit l'image entrée en argument avec la fonction lit_image. Cette fonction récupère le type de l'image, enlève les commentaires, puis appelle alloue_image pour réserver un tableau de taille largeur*largeur*3 permettant d'accueillir les valeurs des pixels de l'image ainsi que les autres tableaux.
Il ne reste plus à la fonction lit_image qu'à remplir un tableau avec le contenu du fichier image puis à le fermer.
On appelle ensuite la fonction nouvelle_image qui crée une nouvelle structure image identique à la structure image précédente.

b) Construction de l'histogramme

Initialiser l'histogramme
L'histogramme est un tableau de 255 valeurs, car au maximum un pixel prend la valeur de 255. On initialise les 3 tableaux de notre image (passés en arguments) qui représentent les quantités de rouge, vert et bleu à 0. Puis nous parcourons le tableau de pixels et à chaque fois nous incrémentons les tableaux en conséquence. Si nous rencontrons un pixel rouge avec pour valeur 50 alors nous faisons image->tabr[50]++.

Créer l'histogramme
On appelle tout d'abord la fonction histogramme. Celle-ci remplit les trois tableaux de la structure image tabr, tabv et tabb de 0, puis incrémente leur valeur en chaque nombre de 0 à 255 à chaque fois qu'elle rencontre cette valeur de couleur dans le tableaux de pixels tabrvb.
Ensuite, la fonction creeHisto crée une structure image qui servira d'histogramme et un tableau d'entiers qu'elle alloue. Elle remplit ensuite ce tableau d'entiers de la moyenne des valeurs des trois tableaux remplis précédemment.
La fonction recherche_max renvoie ensuite le maximum d'intensité du tableau, ce qui permet d'afficher les autres valeurs d'intensité en pourcentage de cette dernière.
Puis on remplit la structure image de l'historique avec ses caractéristiques, et on alloue ses tableaux avec la fonction alloueImage. Il s'agit ensuite simplement de remplir le tableau tabrvb de pixels blancs et noirs afin de créer un histogramme sous forme de diagramme à barres.
Enfin, on écrit l'image avec la fonction ecrire_image, puis on la désalloue et on libère le tableau d'entiers.

Utilisation dans le projet
On appelle les fonctions histogramme et Creehisto qui respectivement crée l'histogramme en mémoire, et l'écrivent puis l'enregistrent sous forme d'un fichier image. On appelle cette fonction au départ pour obtenir l'histogramme de l'image source, puis on l'appellera à chaque fois que on affiche l'image finale pour avoir l'histogramme de celle-ci.

c) Les fonctions du calque

Allouer un calque
La fonction Alloue_calque réserve la place en mémoire pour la structure calque.

Créer le premier calque
La création du premier calque se fait avec le chargement d'une image et l'appllication directe de LUTs, avec la fonction Cree_lut. Elle inverse l'image avec la fonction image_lendroit afin de l'afficher dans le bon sens dans la fenêtre. Elle met ensuite les paramètres d'opacité et d'opération à 1, puis fait pointer les deux pointeurs de la structure vers NULL.

Créer d'autres calques
L'ajout de calques se fait avec la fonction Cree_calque_vierge. Cette fonction alloue une structure calque, avec l'appel à la fonction alloue_calque et y ajoute un LUT vierge pour initialiser la liste. Elle copie les caractéristiques de l'image (largeur, hauteur, etc…) dans la stucture calque, puis remplit les tableaux du calque de blanc.
Enfin, elle ajoute ce nouveau calque à la fin de la liste chaînée en réorientant les pointeurs nécessaires.

Modifier le paramètre d'opacité
On peut modifier le paramètre d'opacité du calque courant avec la fonction Modifie_opacite. On remplace dans la structure calque correspondante le paramètre opacite par celui que l'on a demandé à l'utilisateur.

Modifier la fonction de mélange
On peut modifier la fonction de mélange du calque courant avec la fonction Modifie_fonction. Elle procède à un switch pour modifier le paramètre opération de la structure calque concernée.

Supprimer un calque
On peut supprimer le calque courant. Il s'agit de réorienter les pointeurs des calques précédent et suivant afin que lors du parcours de la liste, le maillon courant ne soit plus atteint.
S'il s'agit du premier calque, on n'agit que sur le pointeur du calque suivant et inversement pour le dernier calque.
S'il n'y a plus qu'un calque, la suppression n'est pas effectuée.
Lors de la suppression d'un calque, le nouveau calque courant devient celui qui se trouve immédiatement en dessous. Si l'on supprime le premier calque, le nouveau calque courant est le nouveau premier calque.
On affiche ensuite dans la fenêtre le nouveau calque courant avec la fonction affiche_image_courante.

Naviguer dans les calques
On peut se placer sur le calque immédiatement à droite ou à gauche, en déplacant le pointeur de la liste sur le calque immédiatement précédent ou suivant : à l'aide de la liste chaînée, on change de calque courant. Si l'on est sur l'un des calques extrêmes, on empêche l'utilisateur de naviguer dans le mauvais sens. On affiche le nouveau calque courant.
On affiche un message qui indique le numéro du calque courant ainsi que le nombre total de calques.

Charger une image sur un calque
On peut charger une image sur le calque courant avec la fonction Charge_image.
Cela consiste à remplir tous les champs de la structure calque avec les caractéristiques de l'image. Puis on appelle la fonction actualise_image pour l'affichage.

Appliquer un LUT à un calque
La fonction applique_lut appelle la fonction insertion_lut, et lui envoie un entier ne fonction du LUT à appliquer. La seconde lui renvoie un LUT que la première ajoute à la fin de la liste de LUT du calque courant.

Afficher un calque
La fonction affiche_image_courante affiche l'image d'un calque avec ses LUTs. Elle parcourt la liste de LUTs du calque courant, en modifiant ses valeurs. Puis elle appelle la fonction actualise_image, qui affiche l'image ainsi crée.

Sortie de l'image finale
A tout moment, on pourra afficher l'image modifiée et la sauvegarder sous le nom image_finale avec la fonction Image_finale.
Elle permet aussi de passer du Mode calque en Mode visualisation.
Si l'on est en Mode calque, cette fonction passe alors le programme en Mode visualisation. Puis elle crée des tableaux de pixels temporaires.
Ensuite, elle parcourt la liste de calques et la liste de LUTs de chaque calque, en modifiant les tableaux. Une fois qu'elle a fini ce parcours, elle affiche la nouvelle image avec la fonction actualiseImage.
Puis elle inverse l'image avec la fonction image_lendroit. Enfin, elle enregistre cette image avec la fonction ecrire_image et crée son histogramme avec la fonction Cree_histo.
Si l'on est en Mode visualisation, on repasse en Mode calque et on affiche le calque courant avec la fonction affiche image courante.
La pression de n'importe quelle touche en Mode visualisation remet le logiciel en Mode calque.

Utilisation dans le projet
Chaque fonction du fichier calque.c a son utilité propre. Il est grâce à elles possible d' initialiser une liste de calques (Cree_calque_fond), d'en ajouter (Cree_calque_vierge), d'en enlever (Supprime_calque), de modifier leurs paramètres (Modifie_opacite, Modifie_operation), de naviguer entre eux (Navigue_calque) tout en bénéficiant d'un aperçu visuel (affiche_image_courante), d'ajouter des LUTs (applique_lut), et enfin d'afficher l'image finale avec ses calques et ses LUTs (Image_finale).

La Sortie du programme
La fonction Sortie entraîne la désallocation des listes de LUTs et des listes de calques. Elle parcourt la liste de calques, et ce faisant, parcourt chaque liste de LUTs en désallouant les structures LUTs, et en appelant la fonction desalloueLookup. Lorsqu'elle a terminé de parcourir la liste de LUT d'un calque, elle desalloue l'image du calque avec la fonction desalloueImage. Puis elle passe au calque suivant et désalloue le calque précédent.
Elle effectue ces opérations pour chaque calque, puis renvoie un message si la désallocation s'est bien passée.

d) Les luts

Allouer un lut
Pour allouer un lut c'est un simple malloc sur la structure lut.

Désallouer un lut
On fait un free sur les luts de l'image.

Créer un lut
La fonction Cree_lut reçoit en argument le type de LUT à créer et la puissance si nécessaire. La fonction crée un LUT qu'elle alloue.
Une différentiation s'opère à ce niveau. Si la valeur de la puissance est différente de 777, alors c'est que la valeur de la puissance a été donnée en ligne de commande. Dans le cas contraire, on la demande à l'utilisateur.
Cette fonction va ensuite envoyer les trois tableaux lookupr, lookupv et lookupb à l'une des fonctions d'application d'effets qui les modifieront en conséquence.

Insérer un lut
La fonction insertion_lut appelle la fonction Cree_lut qui lui renvoie un LUT. La première l'ajoute en fin de liste en réorientant les pointeurs nécessaires.

Supprimer un lut
La suppression d'un lut est en fait la suppression du dernier élément de la liste de lut. On se situe donc en fin de liste sur le dernier lut, puis on recule d'un, on supprime son suivant et on met le suivant du lut courant à null.

Les effets possibles

Effet vierge *:* C'est un effet qui ne change pas les valeurs du pixel. Il a été implémenté pour que chaque calque ait déjà un lut de base, par souci de facilité pour les insertions.

Ajout de luminosité : Pour ajouter de la luminosité on utilise une puissance passé en paramètre pouvant prendre les valeurs de 0 à 255.Cette valeur est ajoutée à la valeur du pixel donnant ainsi une teinte plus clair,lumineuse à l'image car le pixel se rapproche de la valeur 255.

Diminution de luminosité : Pour diminuer de la luminosité on utilise une puissance passé en paramètre pouvant prendre les valeurs de 0 à 255.Cette valeur est soustraite à la valeur du pixel donnant ainsi une teinte plus sombre à l'image car le pixel se rapproche de la valeur 0.

Ajout et diminution de contraste : Cette fonction creuse l'écart entre les couleurs selon une puissance (positive si on ajoute du contraste et négative si on la diminue). Elle ajoute au pixel >127 de la couleur et diminue les pixels<127. On vérifie bien que la valeur ne dépasse pas 255 ou est inférieur à 0, si c'est le cas on la ramène a 255 ou à 0.

Effet négatif : Pour cet effet on soustrait à 255 le pixel on obtient ainsi la pixel inverse. En faisant ainsi à tous les pixels on aura l'inverse de l'image.

Effet sépia : On multiplie par une constante le pixel (0. 0.784 pour le rouge, 0.588 pour le vert et 0.392 pour le bleu). Il faut envoyer à ce lut pas le pixel mais la moyenne des pixels RVB.

Effet bruit : Le bruit est un effet aléatoire. C'est à dire que on n'obtiendra jamais deux fois la même image avec les mêmes paramètres.
Pour réaliser du bruit sur une image, il faut altérer la couleur des pixels. On utilise le paramètre alpha qui représente l'intensité du bruit. On ajoute une valeur comprise entre -alpha/2 et alpha/2 à chaque composante de chaque pixel.

Effet seuil : On regarde si le pixel est <= 127, si c'est el cas alors on le met à 0 sinon on le met a 255.

Effet niveau de gris : De même manière que l'effet sépia on enverra la moyenne de RVB. Les pixels r, v et b prendront la valeur de cette moyenne.

Utilisation dans le projet :
L'ajout d'un LUT se fait avec la fonction applique_lut. Cette fonction envoie à la fonction insertion_lut l'instruction correspondant au LUT à ajouter. La fonction insertion_lut ajoute un LUT à la fin de la liste de LUTs du calque courant. Elle appelle pour cela la fonction Cree_lut. Cette fonction appelle la fonction LUT correspondante, qui modifie les tableaux tabr, tabv et tabb.
La fonction supprime_lut effectue une suppression du dernier élément de la liste de LUTs du calque courant. Elle affiche ensuite l'image sans le LUT qui vient d'être supprimé.

e) L'historique

Initialiser une pile
La fonction iniPile permet d'initialiser l'historique. Elle place le sommet de celle-ci à -1.

Tester si elle est vide
La fonction testVide teste si le sommet de la pile est à -1 ou non, puis renvoie un entier en fonction du cas détecté.

Empiler
La fonction ajoute un élément au sommet de la pile qui contient toutes les informations de l'élément supprimé. Elle est appelée à chaque fois que l'utilisateur effectue une opération de modification, de suppression ou d'ajout.

Dépile
La fonction depile permet d'enlever un élément du sommet de la pile et de le renvoyer.

Affichage de l'historique
La fonction affiche parcourt la pile de l'historique, si elle n'est pas vise, et indique par des messages quel est le type des éléments de la pile.
Sinon, elle indique que l'historique est vide.

Annuler une opération
Si l'historique n'est pas vide, la fonction Annule utilise le retour de la fonction dépile. En fonction du type de la dernière opération effectuée, elle fait l'opération inverse, de manière à annuler cette dernière.
Sinon, elle indique que l'historique est vide.

Utilisation dans le projet
A chaque fois qu'une opération a été réalisée, il place cette opération sur le sommet de la pile à l'aide de la fonction empile. Lorsqu'on demande de voir l'historique des opérations réalisées, la fonction affiche parcourt l'historique et indique la liste des dernières opérations effectuées. On peut demander l'annulation de la dernière opération effectuée à l'aide de la fonction annule.

f) L'IHM

L'application des luts directs
On gère les arguments facultatifs entrés en ligne de commande à l'aide des fonctions lut_direct et lut_supplementaire. Ces arguments ajoutent des LUTs à l'image chargée en ligne de commande.
La fonction lut_direct appelle la fonction Cree_calque_fond. Cette dernière crée une première structure calque et la renvoie. Elle initialise la liste de LUTs en en créant un premier, qui peut être vierge. Elle renvoie ensuite ce calque avec lequel on initialise la liste.
La deuxième appelle la fonction insertion_lut, qui ajoute les LUTS demandés en ligne de commande, en leur transmettant les arguments rentrés en ligne en commande.
La méthode utilisée pour l'ajout des LUTs est ainsi identique à celle qui est employée une fois à l'intérieur du programme, et sera détaillée plus loin.
On initialise alors la liste avec le premier calque crée.

Les fonctions transforme_donnee1 et transforme_donnee2 permettent de récupérer la valeur de la puissance donnée en ligne de commande.

Dans le cas où il n'y a que deux arguments, on appelle Cree_calque_fond et on initialise la liste avec ce calque.

Détection du clavier
La liste a été mis en variable globale pour l'utilisation de l'IHM. En début de programme, la fonction menu_calques active la détection du clavier et lance la fenêtre pour l'affichage.
Les fonctions de détection du clavier récupèrent les entrées et effectuent les actions en conséquence. Toute la suite du programme est donc coordonnée par les deux fonctions de détection du clavier, qui appellent les fonctions correspondant aux entrées qu'elles reçoivent.

Ces fonctions permettent aussi au programme de repasser en Mode calque s'il se trouve en Mode visualisation.

Les modes
Le programme comprend deux modes. Le Mode calque est le mode de travail. Il permet d'effectuer toutes les opérations disponibles dans l'application. Le Mode visualisation affiche l'image avec toutes les modifications qui lui ont été apportées depuis le lancement de l'application.

Affichage inverse pour avoir l'image à l'endroit
La fonction image_lendroit est utilisée pour inverser le sens de l'image et permettre qu'elle s'affiche dans le bon sens dans la fenêtre de visualisation.
Cette fonction échange deux par deux les lignes k et n-k, k variant de 1 à n et n étant le nombre de lignes de pixels sur l'image.

g) Le main

L'initialisation se fait dans le main.
A son entrée, les variables sont crées. L'historique est initialisé et la pile est mise a vide.
Il faut ensuite gérer l'ouverture du fichier en ligne de commande ainsi que les arguments qui appliquent les luts sur le premier calque.
S'il n'y a qu'un argument, on envoie un message d'erreur qui rappelle le format attendu pour le lancement du programme.
S'il y a deux arguments ou plus, on vérifie que le format rentré est adéquat.
On appelle les fonctions creeHisto et histogramme pour créer l'histogramme de l'image de départ.

Conclusion :

Le logiciel de traitement d'images est ainsi conforme au cahier des charges fourni.
Il permet de charger une ou plusieurs images, de travailler sur des calques, d'appliquer des effets sur chacun des calques et de visualiser à tout moment l'image modifiée ainsi que de la sauvegarder sur le disque dur.
Il permet aussi l'affichage d'un historique des opérations effectuées et l'annulation de ces opérations.
Il crée en entrée du programme un histogramme de l'image chargée et en crée un autre à chaque visualisation de l'image modifiée.
Le contrôle de ce programme se fait intégralement au clavier, chaque touche ayant une fonction spécifique.